Image Feature Extraction: GENIE vs Conventional Supervised Classification Techniques

Neal R. Harvey, Steven P. Brumby, Simon Perkins, James Theiler, John J. Szymanski, Jeffrey J.Bloch, Reid B. Porter, Mark Galassi, A. Cody Young,

Abstract— **We have developed an automated feature detection/classification system, called GENIE (GENetic Imagery Exploitation), which has been designed to generate image processing pipelines for a variety of feature detection/classification tasks. GENIE is a hybrid evolutionary algorithm that addresses the general problem of finding features of interest in multi-spectral remotely-sensed images. We describe our system in detail together with experiments involving comparisons of GENIE with several conventional supervised classification techniques, for a number of classification tasks using multi-spectral remotely-sensed imagery.**

Keywords— **Supervised Classification, Genetic Programming, Image Processing, Evolutionary Algorithms, Multispectral Imagery, Remote Sensing.**

I. Introduction

LARGE volumes of remotely-sensed multi-spectral data are being generated from an increasing number of increasingly sophisticated airborne and spaceborne sensor systems. While there is no substitute for a trained analyst, exploitation of this data on a large scale requires the automated extraction of specific features of interest. Creation and development of task-specific feature-detection algorithms is important, yet can be extremely expensive, often requiring a significant investment of time and effort by highly skilled personnel.

Our particular interest is the pixel-by-pixel classification of multi-spectral remotely-sensed images to locate and identify various features of interest. These range from broad-area features such as forest and open water to manmade features such as buildings and roads. The large number of features in which we are interested, together with the variety of instruments with which we work, make the hand-coding of suitable feature-detection algorithms impractical. We are therefore developing a supervised learning approach that can, using only a few hand-classified training images, generate image processing pipelines that are capable of distinguishing features of interest from the background.

In applying general-purpose supervised learning techniques to multi-spectral imagery, the usual approach is to employ purely spectral feature vectors, formed by the set of intensity values in each spectral channel for each pixel in the image. These vectors provide a convenient fixed-dimensionality space in which conventional classifiers can often work well. It is clear, however, that spatial relationships (such as texture, proximity, or shape, all of which are disregarded with purely spectral feature vectors) can be very informative in scene classification. Many different kinds of extra spatial context information could be added to the spectral information, as additional feature dimensions of the pixel feature vector. The problem is that there exists a combinatorically vast choice for these additional feature vector dimensions; yet it is clear that a suitable choice of features could make classification much easier. Unfortunately, this suitable choice is, in general, application-specific.

It is in order to address just this problem that we have developed a hybrid evolutionary algorithm called GENIE (GENetic Imagery Exploitation) [2], [3], [4], [5], [6], [7], [8], that searches through the space of image processing algorithms. GENIE is a hybrid in that the evolutionary part of the program attempts to identify a pipeline of image processing operations which transform the raw multi-spectral data planes into a new set of image planes; these intermediate *feature planes* are then input to a conventional supervised classification technique to provide the final classification results.

The beauty of an evolutionary approach is its flexibility: if we can derive a fitness measure for a particular problem, then it might be possible to solve that problem. Many varied problems have been successfully solved using evolutionary computation, including: optimization of dynamic routing in telecommunications networks [9], optimizing image processing filter parameters for archive film restoration [10], designing protein sequences with desired structures [11] and many others.

When adopting an evolutionary approach, a critical issue is how one should represent candidate solutions in order that they may be effectively manipulated. We use a genetic programming (GP) method of representation of solutions, due to the fact that each individual will represent a possible image processing algorithm. GP has previously been applied to image-processing problems, including: edge detection [12], face recognition [13], image segmentation [14] and feature extraction in remote sensing images [15], [2], [3]. The work of Daida et al. [15], Brumby et al. [2] and Theiler et al. [3] is of particular relevance since it demonstrates that GP can be employed to successfully evolve algorithms for real tasks in remote-sensing applications.

This paper describes our system in detail together with experiments involving comparisons of GENIE with several conventional supervised classification techniques, for a number of classification tasks using multi-spectral remotely-sensed imagery.

The remainder of the paper is organized as follows: Sec-

This research was funded by the U.S. Departments of Energy and Defense

tion II describes the GENIE system in detail. Section III describes the conventional supervised classification techniques with which GENIE is to be compared. Section IV describes the data and classification tasks on which the algorithms are to be tested and compared. Section V describes the results of the comparisons. Section VI discusses the results presented in Section V. A summary and conclusions are presented in Section VII.

II. THE GENIE SYSTEM

GENIE employs a classic evolutionary paradigm: a population of is maintained of candidate solutions (*chromosomes*), each composed of interchangeable parts (*genes*), and each assessed and assigned a scalar fitness value, based on how well it performs the desired task. After fitness determination, the evolutionary operators of selection, crossover and mutation are applied to the population and the entire process of fitness evaluation, selection, crossover and mutation is iterated until some stopping condition is satisfied.

A. Training Data

The environment for each individual in the population consists of *data* planes, each of these planes corresponding to a separate spectral channel in the original image, together with a *weight* plane and a *truth* plane. The weight plane identifies those pixels to be used in training, while the truth plane locates the features of interest in the training data. The data in the weight and truth planes may be derived from actual ground truth (collected on the ground, at or near the time the image was taken) or from the best judgement of an analyst looking at the data. Because collecting ground truth data is so expensive, our system employs a Java-based tool called ALADDIN to assist the analyst in making judgements about and marking out features in the data. Through ALADDIN, the analyst or user can view a multi-spectral image in a variety of ways, and can create training data by painting directly on the image using a computer mouse. Currently, training data consists of binary-valued pixels, with these values being either “true” or “false”. *True* defines areas where the analyst is confident that the feature of interest does exist. *False* defines areas where the analyst is confident that the feature of interest does not exist. Pixels within the training image not labeled as either true or false are not included in the determination of the fitness of an individual. Fig. 1 shows a screen capture of an example session. Here the analyst has marked out golf courses as the particular feature of interest.

B. Encoding Individuals

Each individual *chromosome* in the population consists of a fixed-length string of *genes*. Each gene in GENIE corresponds to a primitive image processing operation. Therefore the entire chromosome describes an algorithm consisting of a sequence of primitive image processing operations [2], [3].

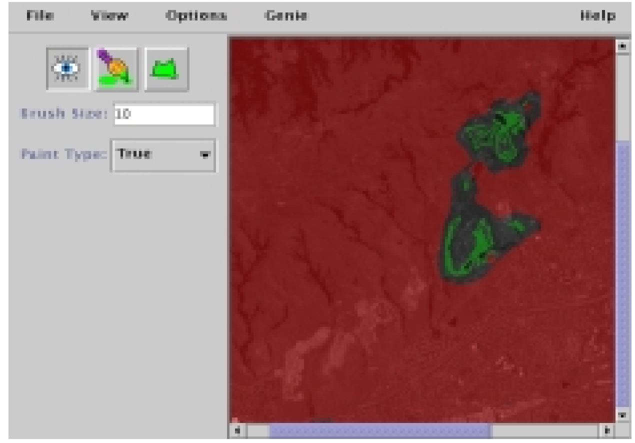


Fig. 1. ALADDIN: the graphical interface used by the analyst to create training data. Note that ALADDIN relies heavily on color, which does not show up well in this image. The light colored patches in the center-right and upper-right parts of the image are two golf courses that have been marked up as “true”. Most of the rest of the image has been marked up as “false”, except for a small region around the golf courses which has been left unmarked.

B.1 Genes and Chromosomes

A single gene consists of an operator name, plus a variable number of input arguments, specifying from where input is to come; output arguments, specifying to where output is to be written; and parameters, modifying the operator’s function. Different operators require different numbers of parameters. Parameters may be integer, floating point, or categorical. The operators used in GENIE take one or more distinct image planes as input, and produce one or more image planes as output. Input can be taken from any data planes in the training data image cube. Output is written to any of a small number of *scratch planes* — temporary workspaces where an image plane can be stored. Genes can also take input from scratch planes, but only if that scratch plane has been written to by another gene positioned earlier in the chromosome sequence.

The image processing algorithm represented by any particular chromosome can be thought of as a directed acyclic graph, where the non-terminal nodes are primitive image processing operations, and the terminal nodes are individual image planes extracted from the multi-spectral image used as input. The scratch planes are the ‘glue’ that combines together primitive operations into image processing pipelines. Traditional GP [16] uses a variable sized (within limits) tree representation for algorithms. Our representation differs in that it allows for re-use of values computed by sub-trees, since many nodes can access the same scratch plane, i.e., the resulting algorithm is a graph rather than a tree. It also differs in that the total number of nodes is fixed, although not all of these may be actually used in the final graph. Currently, crossover is carried out directly on the linear representation.

Our notation for genes is most easily illustrated by an example: the gene **[ADDP rD1 rS1 wS2]** applies pixel-by-pixel addition to two input planes, read from data plane 1 and from scratch plane 1, and writes its output to scratch plane 2. Any additional required operator parameters are listed after the input and output arguments.

Our "gene pool" has been restricted to a set of primitive image processing operators which we consider useful. For different applications, the user may want to choose different sets of primitive operators; for the studies described here, we used all of our available operators. These include spectral, spatial, spatio-spectral, logical and thresholding operators. Table I outlines these operators. For details regarding Laws textural operators, the interested reader is referred to [17], [18].

The set of morphological operators is restricted to function-set processing morphological operators, i.e. grey-scale morphological operators having a flat structuring element. The sizes and shapes of the structuring elements used by these operators is also restricted to a pre-defined set of primitive shapes, which includes, square, circle, diamond, horizontal cross and diagonal cross, and horizontal, diagonal and vertical lines. The shape and size of the structuring element are defined by operator parameters. Other local neighborhood/windowing operators such as mean, median, etc. specify their kernels/windows in a similar way. The spectral operators have been chosen to permit weighted sums, differences and ratios of data and/or scratch planes.

It should be noted that although all chromosomes have the same fixed number of genes, the *effective length* of the resulting algorithm graph may be smaller than this. For example, an operator may write to a scratch plane that is then overwritten by another gene before anything has a chance to read from it. Genie performs an analysis of chromosome graphs when they are created and only carries out those processing steps that actually affect the final result. Therefore, the fixed length of the chromosome acts as a maximum effective length.

C. Backends

Complete (or "hard") classification requires that the algorithm produce a single binary-valued output plane. It would be possible to treat, for example, the final contents of scratch plane **S1** as the output from the algorithm (thresholding of this plane may be required to obtain a binary result). However, such an approach assigns a privileged role to a particular scratch plane label (in this case **S1**), which need not be respected by the algorithm. Therefore we have adopted a hybrid approach which applies a conventional supervised classifier to the set of scratch planes to produce the final answer plane. We have found this approach to be advantageous.

To do this, we first select a subset of the scratch planes and data planes to be *answer planes*. We then apply a more conventional supervised classification technique to these answer planes.

Typically we would use the provided training data and the answer planes to derive the *Fisher Linear Discriminant* [19], which is the linear combination of the answer planes that maximizes the mean separation in spectral terms between those pixels marked up as "true" and those pixels marked up as "false", normalized by the "total variance" in the projection defined by the linear combination. The output of the discriminant-finding phase is a multi-valued (grey-scale) image. This is then reduced to a binary image by finding the threshold value that maximizes the fitness as described in the following section.

D. Fitness Evaluation

The fitness of a candidate solution is given by the degree of agreement between the final binary output plane and the training data. This degree of agreement is determined by the weighted Hamming distance between the final binary output of the algorithm and the training data. In a more formal/mathematical context, if we denote the detection rate (fraction of "feature" pixels classified correctly over the entire scene) as R_d and the false alarm rate (fraction of "non-feature" pixels classified incorrectly over the entire scene) as R_f, and the fitness of a candidate solution as F, then we define the fitness as:

$$F = 500(R_d + (1 - R_f)) \qquad (1)$$

Thus, a fitness of 1000 indicates a "perfect" classification result: i.e. none of those pixels in the training set have been classified incorrectly. This fitness score weights getting all the "feature" pixels correct equivalently to getting all the "non-feature" pixels correct.

E. Software Implementation

The evolutionary algorithm code has been implemented in object-oriented Perl. This provides a convenient environment for the string manipulations required by the evolutionary operations and simple access to the underlying operating system (Linux). Chromosome fitness evaluation is the computationally intensive part of the evolutionary process and we currently farm this job out to a separate process running RSI's IDL [20] language and image processing engine. IDL currently does not have all the image processing operators of interest to us, so we have implemented additional operators in C that can be called from within the IDL environment. Within IDL, individual genes correspond to single primitive image operators, which are coded as IDL procedures, with a chromosome representation being coded as an IDL batch executable. In our present implementation, an IDL session is opened at the start of a run and communicates with the Perl code via a two-way UNIX pipe. This pipe is a low-bandwidth connection. It is only the IDL session that needs to access the input and training data (possibly hundreds of Megabytes), requiring a high-bandwidth connection. The Aladdin training data mark-up tool was written in Java. Fig. 2 shows the software architecture of the system.

III. Conventional Supervised Classification

Many implementations of standard supervised classifiers exist. One of the most widely used remote-sensing software packages is the ENvironment for Visualizing Imagery

Gene Abbreviation	Image Processing Operation	Inputs/ Outputs/ Params	Notes
ADDP	Add planes	2/1/0	Basic mathematical operations. ADDS adds a scalar, which may be negative, to its input. DIFF is like SUBP but outputs the absolute values. NDI is like SUBP, but divides the result by the sum of its two inputs. MULTS scales its input by a scalar, which by default is positive. LINSCL is like MULTS but takes an extra param which is added onto the scaled input. LINCOMB outputs a linear combination of its two inputs, in proportion specified by its one parameter, which takes a value between 0 and 1.
ADDS	Add scalar	1/1/1	
SUBP	Subtract planes	2/1/0	
DIFF	Absolute difference	2/1/0	
NDI	Normalized difference	2/1/0	
MULTS	Multiply by scalar	1/1/1	
NEG	Negate plane	1/1/0	
MULTP	Multiply planes	2/1/0	
SQRT	Square root	1/1/0	
SQR	Square	1/1/0	
LINSCL	Linear scale	1/1/2	
LINCOMB	Linear combination	2/1/1	
MIN	Minimum	2/1/0	Logical operations. MIN and MAX perform pixel-wise minimum and maximum, equivalent to AND and OR for binary input. IFLTE outputs its third input wherever the first input is less than its second input, and its fourth output elsewhere.
MAX	Maximum	2/1/0	
IFLTE	‘If less than else’	4/1/0	
CLIP_HI	Clip high	1/1/1	Thresholding operations. CLIP_HI truncates any pixel values above a value set by its param. CLIP_LO does the converse. THRESH sets all values below its threshold param to 0, and all those above to `dataScale`.
CLIP_LO	Clip low	1/1/1	
THRESH	Threshold	1/1/1	
SAVAR	Spectral angle variance	2-16/1/2	Spectral angle operations. SAVAR and SADIST look at two circular neighborhood regions around each pixel, of size defined by their two params. SAVAR returns the difference between the variance of the spectral angles of the pixels in the two regions. SADIF returns the difference between the mean spectral angle of both regions. SADIST returns the spectral angle difference between each pixel and the vector defined by its params. SANORM normalizes the vector defined by its inputs to have a magnitude equal to `dataScale`.
SADIF	Spectral angle difference	2-16/1/2	
SADIST	Spectral angle distance	2-10/1/2-10	
SANORM	Normalize spectral vector	2-10/2-10/0	
QTREG	Region Size related to Statistics	1/3/1	QTREG Determines the region size (in log base 2) around each pixel for which the normalized variance per pixel standard of the square region first reaches a given threshold. Also returns planes with the linear fit slope and offset of the variance as a function of region scale for each pixel
R5R5	Law’s texture measure	1/1/0	Neighborhood operations. In general, all these operations take a single plane as input and produce a single output plane. The output at each pixel is determined by looking at the pixel’s neighborhood. R5R5, LAWB, LAWD, LAWF and LAWH are widely-used texture measures, developed by Law, that return zero if the neighborhood contains all the same value of pixel, and some other value otherwise, depending upon the distribution of pixel values. R5R5 is corresponds to Law’s $R5^T \times R5$ 5×5 operator. The others are 3×3 operators, corresponding to Law’s $S3^T \times L3$, $E3^T \times E3$, $L3^T \times S3$ and $S3^T \times S3$ operators respectively. Most of the other operators are familiar image processing or morphological operators, whose description can be found in any good book on image processing. Most take two parameters which give the size and shape of a structuring element defining the neighborhood to which the operator is applied. ASF stands for ‘Alternating Sequential Filter’. MB_EDGE takes an additional parameter defining a threshold for edge strength to be looked for. The single parameter for H_DOME and H_BASIN defines the pixel value offset used by these operators.
LAWB	Law’s texture measure	1/1/0	
LAWD	Law’s texture measure	1/1/0	
LAWF	Law’s texture measure	1/1/0	
LAWH	Law’s texture measure	1/1/0	
LAPLAC3	3x3 Laplacian	1/1/0	
LAPLAC5	5x5 Laplacian	1/1/0	
MORPH_LAPLAC	Morph. Laplacian	1/1/2	
ISO_GRAD	Isotropic gradient	1/1/0	
MEAN	Mean	1/1/1	
VARIANCE	Variance	1/1/2	
SKEWNESS	Skewness	1/1/2	
KURTOSIS	Kurtosis	1/1/2	
SKEW_COEFF	Skewness coefficient	1/1/2	
KURT_COEFF	Kurtosis coefficient	1/1/2	
SD	Standard deviation	1/1/2	
RANGE	Morphological Gradient	1/1/2	
MEDIAN	Median	1/1/2	
EROD	Erode	1/1/2	
DIL	Dilate	1/1/2	
OPEN	Open	1/1/2	
CLOS	Close	1/1/2	
OPCL	Open-close	1/1/2	
CLOP	Close-open	1/1/2	
ASF_CLOP	ASF Close-open	1/1/2	
ASF_OPCL	ASF Open-close	1/1/2	
POS_TH	Positive top hat	1/1/2	
NEG_TH	Negative top hat	1/1/2	
OP_REC	Open with reconstruction	1/1/2	
CL_REC	Close with reconstruction	1/1/2	
H_DOME	H-dome	1/1/1	
H_BASIN	H-basin	1/1/1	
MB_EDGE	Canny edge detector	1/1/2	

TABLE I

The primitive image processing operators (genes) used in Genie and what they do. Space precludes a complete description of the details of the more complex operators, but this table gives the general picture of what kinds of operators are in Genie.

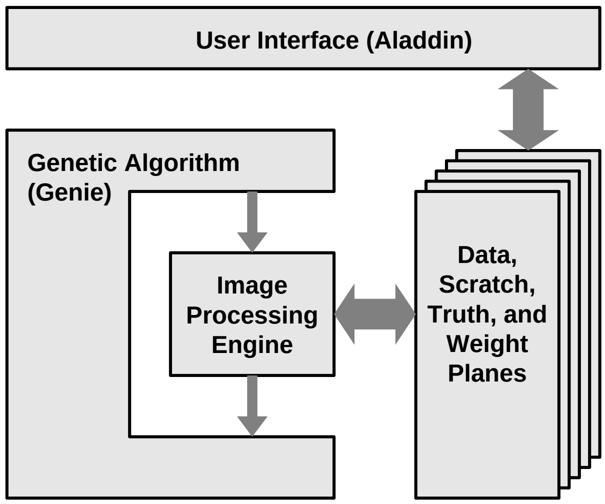


Fig. 2. Software Architecture of the System.

(ENVI) [1]. Supervised classification techniques provided as part of the ENVI package were used in the comparison experiments with Genie. Currently Genie is set up to be able to classify every pixel in its input data into one of two classes: "feature" and "non-feature". The normal mode of operation of the ENVI supervised classifiers is to use training data for the one "true" class: i.e. the feature of interest. The ENVI classifier is then used to classify the input image into "feature" or "unclassified". The user adjusts the parameters of the particular supervised classifier in order to attain optimal performance, with respect to feature identification. The one exception to this is the Maximum Likelihood classifier, where it is necessary to have more than one class in the training data. In this case we used the "feature" and "non-feature" classes and the Maximum Likelihood classifier classified every pixel in the input data into one or other of these two classes, with no "unclassified" pixels being allowed. For applying the ENVI-supplied classifiers to out-of-training-sample data, the training data (reference spectra) used in the training was provided, together with the parameters that gave optimal performance on the training data. For the GENIE case, it was simply a case of applying the algorithms found by GENIE to the out-of-training-sample data (including the linear discriminant and threshold found during training).

Supervised classification is used to categorize data into classes corresponding to user-defined training classes. The training classes for the ENVI supervised classifiers were obtained from the user-defined training data produced using Aladdin. The "truth" and "weight" planes were read into ENVI and used in the definition of suitable regions of interest for the required training classes: "true" and/or "false", to be used in the subsequent supervised classification.

The following ENVI-supplied supervised classification techniques were used in the comparison experiments:

- Minimum distance
- Maximum likelihood
- Mahalanobis distance
- Spectral angle mapper
- Binary encoding

A. Minimum Distance

The minimum distance supervised classification technique [21], [22] calculates the mean pixel vector of the "feature" class. It assigns new pixels to the "feature" class if the Euclidean distance from that pixel to the mean is less than a user-defined threshold, or to "non-feature" otherwise.

B. Maximum Likelihood

Maximum likelihood classification [21], [22] is the most common supervised classification method used with remote sensing data (as stated in [22]). The maximum likelihood classifier models the distributions of the "feature" and "non-feature" classes as separate multivariate normal distributions. New pixels are assigned to the class that had the highest probability of generating that pixel.

C. Mahalanobis Distance

The Mahalanobis distance technique [21], [22] is very similar to the maximum likelihood classifier, but with the simplification that all classes are modelled as having identical covariance matrices (which defines the shape and orientation of the normal distribution). In the one class case, we compare the probability that a new pixel was generated by the "feature" class, to a user-defined threshold, in order to decide to which class that pixel belongs.

D. Spectral Angle Mapper

The spectral angle mapper (SAM) technique [21], [23] is motivated by the observation that changes in illumination caused by shadows, slope variation, sun position, light cloud, etc., approximately only alter the magnitude of a pixel's vector, rather than the direction. Therefore we can eliminate these effects by normalizing all pixel vectors to unit magnitude and then looking at the angle between a given pixel and the mean vector for the "feature" class. Pixels are assigned to the "feature" class if this angle is less than a user-defined threshold.

E. Binary Encoding

Binary encoding classification [21], [22], [24] encodes the data and reference spectra into ones and zeros, based on whether a particular band value lies above or below the spectrum mean. The comparison between the encoded reference spectrum with the encoded data spectra is performed using a Boolean logic exclusive OR function. A user specifies the minimum fraction of bands that must match between the encoded reference spectrum and the data spectra. Pixels that do not meet this criterion are labeled as "non-feature".

IV. Experimental Data and Classification Tasks

A. Data Used in the Experiments

The remotely-sensed images referred to in this paper were derived from the Airborne Visible and InfraRed Imaging Spectrometer (AVIRIS) [25], a sensor developed and operated by the NASA Jet Propulsion Laboratory. The AVIRIS sensor collects data in 224 contiguous, relatively narrow (10 nm), uniformly-spaced spectral channels. AVIRIS is an airborne sensor and spatial resolution can vary from a few meters to 20 meters. The studies reported here use a reduced number of relatively wide spectral bands. Several of the authors are involved with a new remote sensing satellite called the Multispectral Thermal Imager (MTI) [26]. The MTI satellite was launched in March 2000 and collects data in 15 spectral bands. Ten of these bands sample wavelengths between 0.4 and 2.4 microns, a region covered by the AVIRIS instrument. As test data to develop analysis codes, AVIRIS data were convolved with the MTI spectral filter functions to produce simulated MTI data. This 10-band simulated data was used for development of both conventional remote sensing algorithms and for Genie development.

The images displayed here are false-color images (which have then been converted to grey-scale in the printing process). The color mappings used are the same for all original image data shown. The particular color mappings used here involve averaging MTI bands A (0.45–0.52 μm) and B (0.52–0.60 μm) for the blue component, bands C (0.62–0.68 μm) and D (0.76–0.86 μm) for the green component and bands E (0.86–0.89 μm) and F (0.91–0.97 μm) for the red component. In addition, the images have been contrast enhanced. The choice of color mappings was arbitrary, in that it was a personal decision made by the analyst, made in order to best "highlight" the feature of interest, from his/her perspective and thus enable him/her to provide the best possible training data. This choice of color-mappings, together with a contrast-enhancement tool, are important and very useful features of Aladdin.

B. Classification Tasks

We chose four different features of interest:

- Roads
- Golf Courses
- Urban Areas
- Clouds

These features were chosen because of their particular attributes in multi-spectral data. The features were considered a good test of a supervised classification technique due to the different levels of difficulty they posed for these techniques. Clouds are relatively easy, and mostly spectral; urban areas encompass a land-cover distinction; roads are easy for the eye to find, but notoriously difficult for automated algorithms; golf courses require a combination of spectral and spatial information to disambiguate them from other similarly-vegetated areas (e.g. lawns).

We set the various supervised classification techniques the task of distinguishing these features within several scenes of the 10-channel multi-spectral data as described above. For each feature of interest three separate scenes had training data marked-up using the Aladdin tool. This provided "ground truth" for training data and for assessing the performance of the classification scheme on out-of-training-sample data. We employed a cross-validation scheme where, for each feature, we trained a classifier separately on the three marked-up scenes, and then for each scene, applied the resulting classifier to the two remaining out-of-sample scenes. Genie was run, with a population of 100 individuals, for 500 generations, or until a (perfect score) fitness of 1000 was achieved.

An example of an image plus associated training data is shown in Fig. 3. This figure shows the false-color image for one of the scenes used for the "urban area" feature classification, and the associated training data. In the training data image the white pixels correspond to the places on the image where the feature is asserted to be, the grey pixels to where the feature is asserted not to be, and the black pixels correspond to places where no assertion is made.

V. Comparison Experiments

For the training phase, we ran GENIE and the ENVI-supplied classifiers on the training data. For GENIE, the result of this training phase is an image processing pipeline which can be applied to and tested on other data. To apply the ENVI-supplied classifiers to out-of-training-sample data it was necessary to save the regions of interest of the marked-up training classes and provide them as the reference spectra for application of the classifiers to out-of-training-sample data.

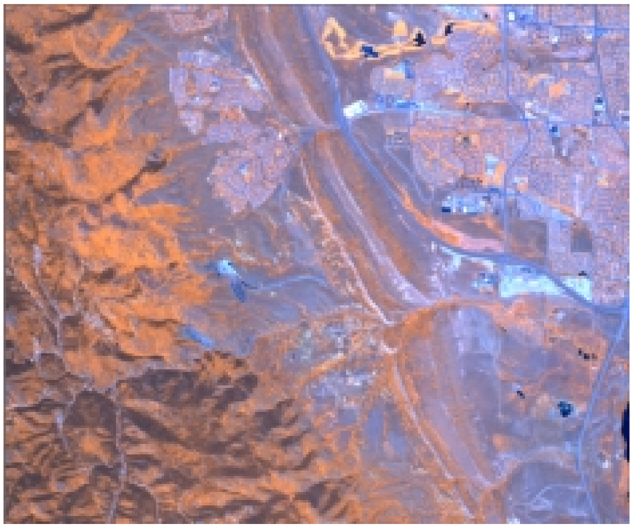
For an objective comparison we measured the fitness, detection rate and false-alarm rate of all the classifiers on the training data and out-of-training-sample data.

Due to the limitations of space we cannot show all the qualitative and quantitative results for all the experiments undertaken. However, we can provide the quantitative results, in terms of the average performance of the different classification algorithms for each feature sought for all classification tasks set.

Table II shows the quantitative results of the comparison between the Genie algorithm output and the ENVI algorithms' output on the training data for all the features. The bottom 3 rows of the table show the average, for each classification technique, across all features sought.

Table III shows the quantitative results of the comparison between the Genie algorithm output and the ENVI algorithms' output on the out-of-training-sample data for all the features. Again, the last 3 rows of the table show the average, for each classification technique, across all features sought.

Another interesting and useful way of presenting the data in Tables II and III is to show the ranking of the various techniques, with respect to fitness, for the various features sought. Table IV shows the ranking of the supervised classification algorithms for the training data, for each feature



(a) (b)

Fig. 3. (a) False-color images of one of the scenes used to produce the training data for "Urban Areas" (urban1) (b) Training data provided for the training scene for "Urban Areas" (White = Feature, Grey = Not Feature, Black = No Assertion)

		GENIE	Min. Dist.	Mahal. Dist.	Max. Likeli.	S.A.M.	Bin. Enc.
Roads	Fitness	963.3	781.2	865.9	921.1	803.9	677.5
	D.R. (%)	96.61	78.63	83.11	91.62	82.93	82.00
	F.A.R. (%)	3.95	22.40	10.03	7.40	22.14	46.50
Golf	Fitness	998.3	945.0	947.6	966.2	915.4	820.1
	D.R. (%)	99.68	95.16	93.84	96.21	92.73	78.72
	F.A.R. (%)	0.00	6.17	4.32	2.97	9.64	14.71
Urban	Fitness	998.9	694.7	861.6	963.6	636.3	580.4
	D.R. (%)	99.85	58.55	80.34	95.67	75.03	83.59
	F.A.R. (%)	0.07	19.61	8.03	2.94	47.77	67.52
Clouds	Fitness	999.9	975.7	946.7	997.9	979.4	760.2
	D.R. (%)	99.99	96.41	94.21	99.91	98.18	55.59
	F.A.R. (%)	0.00	1.27	4.86	0.33	2.29	3.55
Average	Fitness	990.1	849.1	905.5	962.2	833.8	709.5
	D.R. (%)	99.03	82.19	87.87	95.86	87.22	74.98
	F.A.R. (%)	1.01	12.36	6.81	3.41	20.46	33.07

TABLE II
Comparison of Genie's Evolved Algorithm with ENVI Algorithms, on training data (D.R. = Detection Rate, F.A.R. = False Alarm Rate)

sought. Table V shows the overall ranking for the classification algorithms, averaged over all the features for the training data. Table VI shows the ranking for the out-of-training-sample data, for each feature. Table VII shows the overall ranking for the classification algorithms, averaged over all the features for the out-of-training-sample data.

In order to illustrate the results of these classification techniques on training and out-of-training-sample data, the following images are provided (Figs. 4, 5, 6, 7). Again, limitations of space preclude us from providing a complete set of images for every experiment. Here we show an example of Genie and the best-performing ENVI classifier, applied to a training set, and an example of Genie and the best-performing ENVI classifier applied to some out-of-training-sample data.

The "short" (redundant genes stripped out) version of the chromosome (image processing pipeline) found by Genie for the golf course-finding task described above is detailed below:

[QTREG rD1 wS5 wS3 wS1 0.05][LAWC rD7 wS2][MEAN rS2 wS2 3 0][VARIANCE rD7 wS4 3 0][CLOP rS2 wS2 3 0][RANGE rD10 wS1 3 0][OP_REC rS4 wS4 3 0][ASF_OPCL rD2 wS3 3 0]

The graphical representation of this pipeline is illustrated in Fig. 8. Note that the circled **D**s represent the input data planes and the circled **S**s represent the answer planes that are input to the back-end classifier (Fisher Linear Discriminant plus thresholding), to produce the final classification

		GENIE	Min. Dist.	Mahal. Dist.	Max. Likeli.	S.A.M.	Bin. Enc.
Roads	Fitness	763.2	559.9	500.0	611.2	566.0	587.1
	D.R. (%)	60.71	27.54	0.00	62.25	30.29	72.94
	F.A.R. (%)	7.36	15.57	0.00	40.02	17.09	55.51
Golf	Fitness	739.8	584.8	500.0	553.1	696.5	572.9
	D.R. (%)	61.60	42.34	0.00	10.93	58.76	51.16
	F.A.R. (%)	13.65	25.39	0.00	0.31	19.46	36.57
Urban	Fitness	813.5	586.2	514.6	569.4	499.1	521.9
	D.R. (%)	66.32	27.36	2.93	65.86	50.67	70.18
	F.A.R. (%)	3.63	10.11	0.02	51.97	50.51	65.80
Clouds	Fitness	978.0	968.6	632.0	701.7	975.3	727.1
	D.R. (%)	97.43	95.38	28.62	99.97	97.28	48.85
	F.A.R. (%)	1.82	1.67	2.22	59.64	2.23	3.43
Average	Fitness	823.6	674.9	536.6	608.8	684.2	602.3
	D.R. (%)	71.51	48.16	7.89	59.75	59.25	60.78
	F.A.R. (%)	6.62	13.18	0.56	37.99	22.32	40.33

TABLE III
COMPARISON OF GENIE'S EVOLVED ALGORITHM WITH ENVI ALGORITHMS, ON OUT-OF-TRAINING-SAMPLE DATA (D.R. = DETECTION RATE, F.A.R. = FALSE ALARM RATE)

	Roads		Golf		Urban		Clouds	
Rank	Classifier	Fitness	Classifier	Fitness	Classifier	Fitness	Classifier	Fitness
1st	GENIE	963.3	GENIE	998.3	GENIE	998.9	GENIE	999.9
2nd	Max. Likeli.	921.2	Max. Likeli.	966.2	Max. Likeli.	963.6	Max. Likeli.	997.9
3rd	Mahal. Dist.	865.9	Mahal. Dist.	947.6	Mahal. Dist.	861.6	S.A.M.	979.4
4th	S.A.M.	803.9	Min. Dist.	945.0	Min. Dist.	694.7	Min. Dist.	975.7
5th	Min. Dist.	781.2	S.A.M.	915.4	S.A.M.	636.3	Mahal. Dist.	946.7
6th	Bin. Encoding	677.5	Bin. Encoding	820.1	Bin. Encoding	580.4	Bin. Encoding	760.2

TABLE IV
RANKING, WITH RESPECT TO FITNESS, OF SUPERVISED CLASSIFICATION ALGORITHMS, ON TRAINING DATA

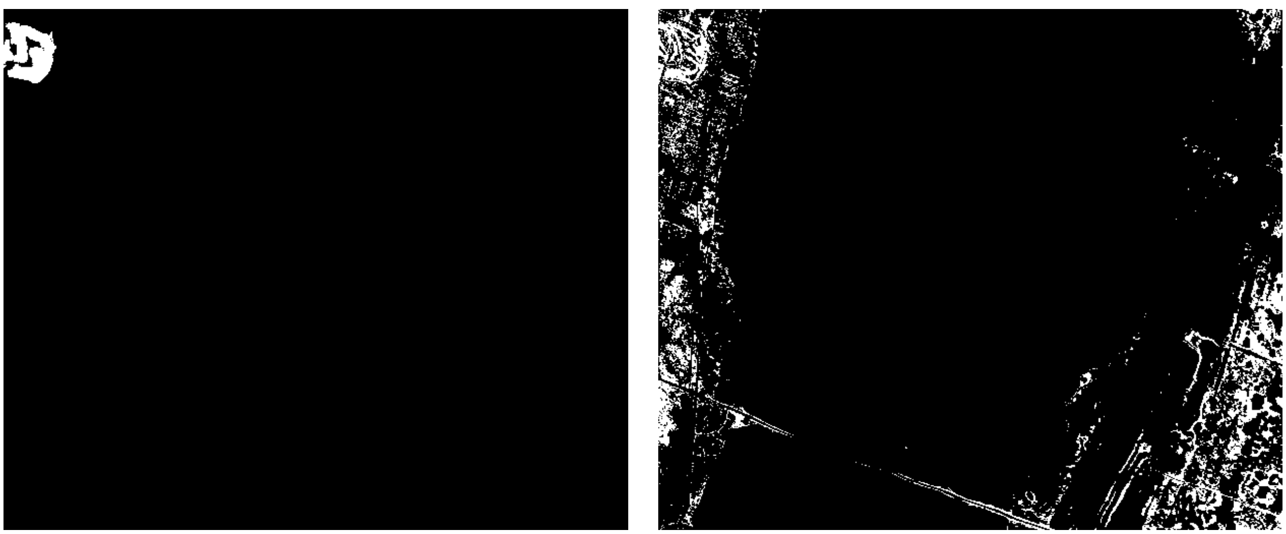
	Roads		Golf		Urban		Clouds	
Rank	Classifier	Fitness	Classifier	Fitness	Classifier	Fitness	Classifier	Fitness
1st	GENIE	763.2	GENIE	739.8	GENIE	813.5	GENIE	978.0
2nd	Max. Likeli.	611.2	S.A.M.	696.5	Min. Dist.	586.2	S.A.M.	975.3
3rd	Bin. Encoding	587.1	Min. Dist.	584.8	Max. Likeli.	569.4	Min. Dist.	968.6
4th	S.A.M.	566.0	Bin. Encoding	572.9	Bin. Encoding	521.9	Bin. Encoding	727.1
5th	Min. Dist.	559.9	Max. Likeli.	553.1	Mahal. Dist.	514.6	Max. Likeli.	701.7
6th	Mahal. Dist.	500.0	Mahal. Dist.	820.1	S.A.M.	499.1	Mahal. Dist.	632.0

TABLE VI
RANKING, WITH RESPECT TO FITNESS, OF SUPERVISED CLASSIFICATION ALGORITHMS, ON OUT-OF-TRAINING-SAMPLE DATA

result. It can be seen that this image processing pipeline has only used 4 of the available 10 data planes as input: data planes D2, D4, D7 and D10. These correspond to the MTI bands B (0.52–0.60 μm: Green), D (0.76–0.86 μm: NIR), G (0.99–1.04 μm: NIR) and O (2.08–2.35 μm: SWIR) respectively. GENIE's choice of input data bands is not surprising, given the task. The algorithm is using the green band, as well as two near-infrared bands and a short-wave infra-red band. Vegetation is known to be evident in the two near-infrared bands GENIE selected and it is well known that golf courses are usually quite green. GENIE produced a solution with five answer planes, and the back-end produced a linear combination of those planes, along with a threshold value, to give a binary classification. Of these five answer planes the most important were S1, S2 and S4; using only those planes we could still achieve the same fitness value, on the training data, as when all the answer planes were used. For the out-of-training-sample

(a) (b)

Fig. 4. (a) False-color images of one of the scenes used to produce the training data for "Golf Courses" (golf3) (b) Training data provided for the training scene for "Golf Courses" (White = Feature, Grey = Not Feature, Black = No Assertion)



(a) (b)

Fig. 5. (a) GENIE results on training data: Fitness = 999.2 (b) Best ENVI classifier for the particular training scene (Minimum Distance): Fitness = 957.4

case, using these 3 answer planes instead of all five also produced comparable performances to when all 5 answer planes were used. It is therefore fair to say that answer planes S3 and S5, in this case, perform no useful function. The outputs of these useful answer planes, as can be seen from Fig. 8, are derived from the NIR and SWIR bands. In this case we see that, perhaps surprisingly, the green band is actually not contributing anything useful.

In an interesting parallel to "junk DNA" in natural chromosomes, the final chromosomes produced by GENIE often exhibit some redundancy, i.e. genes and answer planes that do not significantly contribute to the answer. While these "junk genes" do not affect the functionality of the chromosome, they can make it harder to understand how the chromosome works. We have therefore developed a simple post-run pruning process that removes junk genes and ineffective answer planes from the final solution if this is required.

VI. Discussion

GENIE out-performed all the other classification techniques on all features on both training data and out-of-training-sample data. For the training data, the gap, with respect to fitness, between GENIE's performance and the best of the other techniques was much less than for the out-of-training-sample case. This suggests that GENIE is significantly better at generalizing than the other techniques compared here. An interesting observation is that the best of the other techniques on the training data did not necessarily guarantee it to be the best of the other techniques on the out-of-training-sample data. This indicates the sensitivity of these techniques to training data and confirms

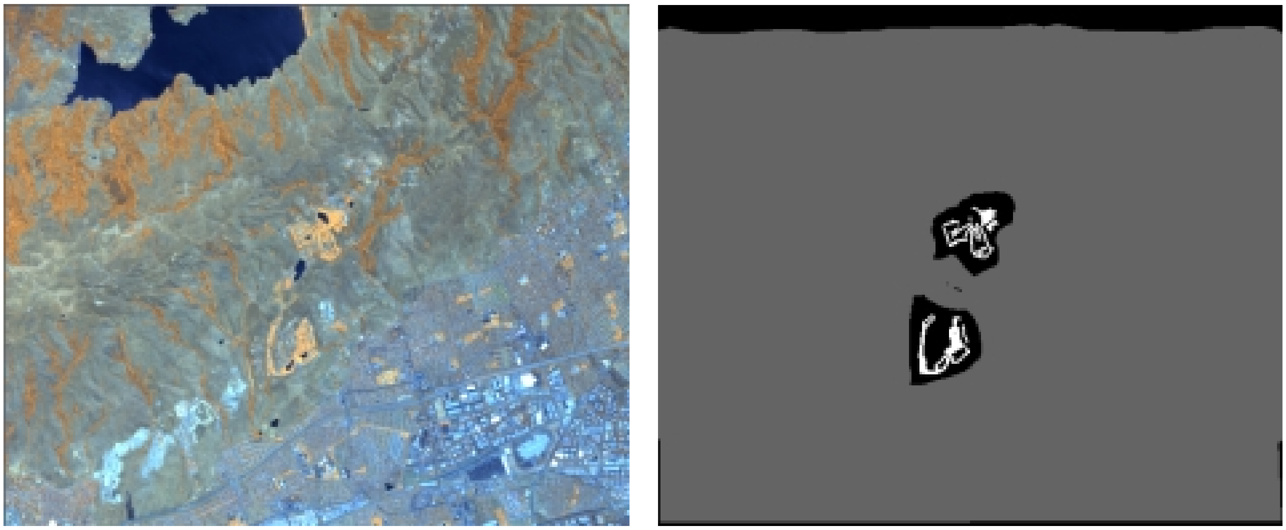

(a) (b)

Fig. 6. (a) False-color images of one of the scenes used to produce training data for "Golf Courses" (golf1) (b) Training data provided for the training scene for "Golf Courses" (White = Feature, Grey = Not Feature, Black = No Assertion)

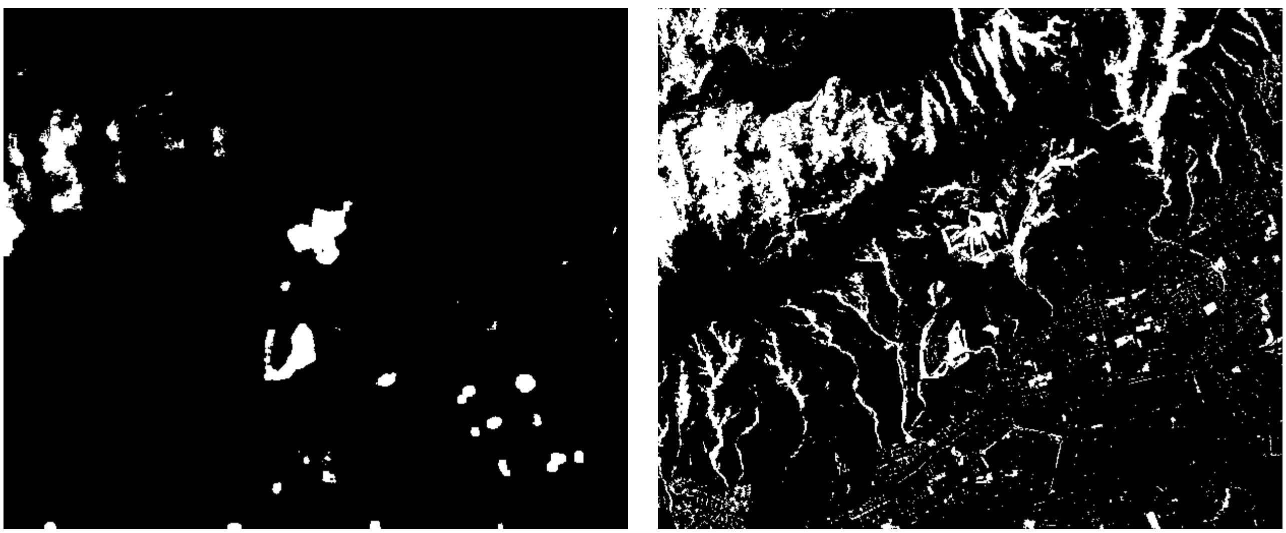

(a) (b)

Fig. 7. (a) GENIE results on out-of-training-sample data: Fitness = 946.9 (b) Best ENVI classifier (for particular training scene) on out-of-training-sample data (Spectral Angle Mapper): Fitness = 856.7

Genie's superior generalization abilities.

One issue to be addressed is training time. At present Genie requires the testing of potentially thousands of candidate algorithms on the training data. Depending on the size of the data, this can take hours to complete. Therefore, Genie can take considerably longer to train than the other techniques. It should be noted, though, that the result of Genie's training is an image processing algorithm that can be applied to other data with times comparable to that of the other techniques' application to out-of-training-sample data. If training time is of great importance a user would perhaps have to weigh up the potentially longer training times of Genie against its potentially far better performance. A few hours is really a small price to pay for a feature identification algorithm that is customized not only to the specific feature, but also to the specific data set. Another point to consider is that being a population-based optimization technique, Genie lends itself well to parallelization, which can dramatically reduce training time. Some experiments have been carried out to demonstrate this [6].

Although traditional classification techniques, as compared here, only use spectral information, it is possible to enable these techniques to use spatial information as well. This can be done by applying some spatial operators to each plane in the input multispectral data and then combining these new processed data planes with the raw data planes and providing both as input to the supervised classifiers. We undertook some experiments where we applied a number of morphological smoothings at different scales to the input data and combined this with the original data. What we found was that this information did help the conventional supervised classifiers in achieving superior per-

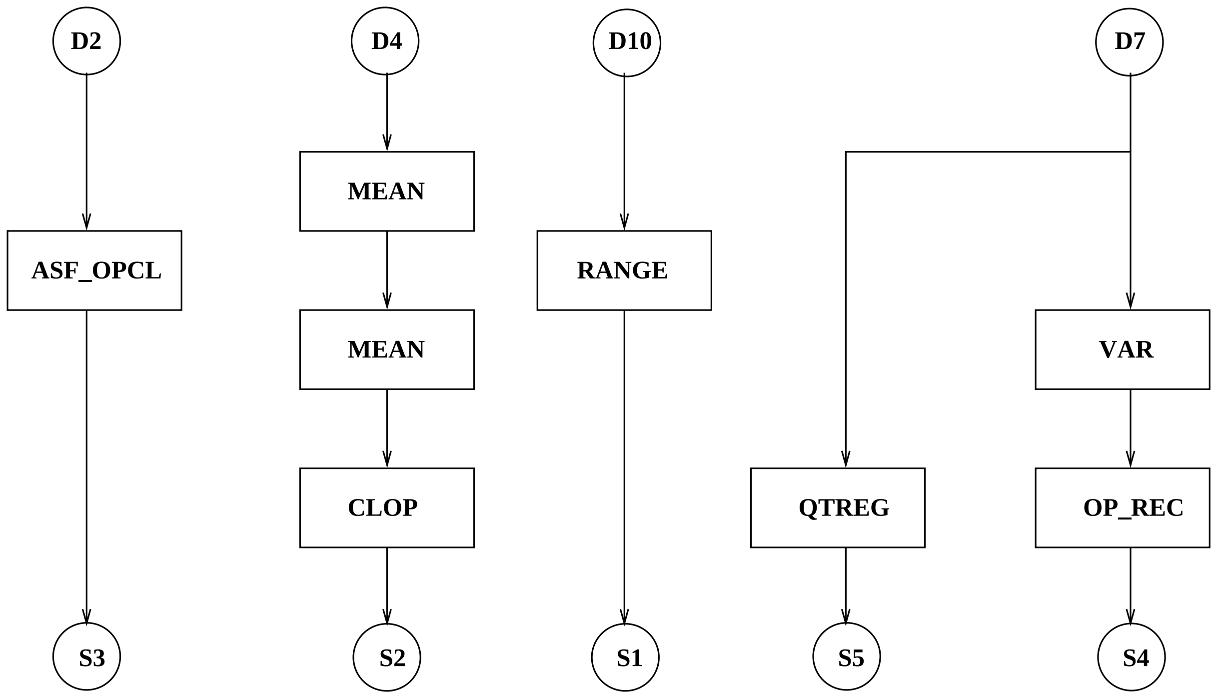


Fig. 8. Image processing pipeline discovered by GENIE for finding golf courses

Rank	Classifier	Fitness
1st	GENIE	990.1
2nd	Max. Likeli.	962.2
3rd	Mahal. Dist.	905.5
4th	Min. Dist.	849.1
5th	S.A.M.	833.8
6th	Bin. Encoding	709.5

TABLE V

Overall ranking, with respect to fitness, of supervised classification algorithms, on training data

Rank	Classifier	Fitness
1st	GENIE	823.6
2nd	S.A.M.	684.2
3rd	Min. Dist.	674.9
4th	Max. Likeli.	608.9
5th	Bin. Encoding	602.3
6th	Mahal. Dist.	536.7

TABLE VII

Overall ranking, with respect to fitness, of supervised classification algorithms, on out-of-training-sample data

formance compared to the same classifiers applied to just the raw data. However, it was still considerably below the performance of GENIE on the original data. Also, the improved performance attained was only on the training data. The classifiers actually performed worse on out-of-training-sample data (i.e., they were less robust). Obviously, if one were to adopt this approach, the choice of which spatial operators to apply is very important and the search space in this regard is immense. If one considers a scenario where some sophisticated technique is used to search the space for the optimal combination of spatial operators, one is entering the arena in which GENIE is designed to function.

VII. Conclusions

An automated feature detection/classification system has been described. In experiments designed to compare GENIE's performance with traditional supervised techniques, GENIE shows better performance than all the other techniques studied here, on both training data and out-of-training-sample data. A great deal of GENIE's superiority can be attributed to the fact that it is capable of very naturally combining information from both the spectral and spatial domain, while the more traditional techniques do not provide a good way to make those combinations.

References

[1] http://www.rsinc.com/envi/index.cfm

[2] S.P. Brumby, J. Theiler, S.J. Perkins, N.R. Harvey, J.J. Szymanski, J.J. Bloch, M. Mitchell, "Investigation of Image Feature Extraction by a Genetic Algorithm", *Proc. SPIE 3812 (1999)*, pp. 24–31.

[3] J. Theiler, N.R. Harvey, S.P. Brumby, J.J. Szymanski, S. Alferink, S. Perkins, R. Porter, J.J. Bloch, "Evolving Retrieval Algorithms with a Genetic Programming Scheme" *Proc. SPIE 3753 (1999)*, pp. 416–425.

[4] N.R. Harvey, S. Perkins, S.P. Brumby, J. Theiler, R.B. Porter, A.C. Young, A.K. Varghese, J.J. Szymanski, J.J. Bloch, "Finding Golf Courses: The Ultra High Tech Approach", *Proc. EvoIASP2000: The Second European Workshop on Evolution-*

ary Computation in Image Analysis and Signal Processing Edinburgh, United Kingdom, April 17, 2000.

[5] S.P. Brumby, J.P. Theiler, N.R. Harvey, S.J. Perkins, R.B. Porter, J.J. Szymanski, J.J. Bloch, "'Genetic Algorithms for Combining New and Existing Tools for Multispectral Imagery', *Proc. SPIE's 14th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls (AEROSENSE 2000)*, Orlando, Florida, April 24-28, 2000.

[6] N.R. Harvey, S.P. Brumby, S.J. Perkins, R.B. Porter, J. Theiler, A.C. Young, J.J. Szymanski, J.J. Bloch, "Parallel Evolution of Image Processing Tools for Multispectral Imagery", *Proc. SPIE 4132, (2000)*.

[7] K.L. Hirsch, S.P. Brumby, N.R. Harvey, A.B. Davis, "The MTI Dense-Cloud Mask Algorithm Compared to a Cloud Mask Evolved by a Genetic Algorithm and to the MODIS Cloud Mask", *Proc. SPIE 4132, (2000)*.

[8] S.J. Perkins, J. Theiler, S.P. Brumby, N.R. Harvey, R.B. Porter, J.J. Szymanski, J.J. Bloch, "GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multispectral Images", *Proc. SPIE 4120, (2000)*.

[9] L.A. Cox Jr., L. Davis, Y. Qiu, "Dynamic anticipatory routing in circuit-switched telecommunications networks", in *Handbook of Genetic Algorithms*, L. Davis (Ed.), pp. 124–143, Van Nostrand Reinhold, New York, 1991.

[10] N.R. Harvey, S. Marshall, "GA Optimization of Multidimensional Grey-Scale Soft Morphological Filters with Applications in Archive Film Restoration", in *Mathematical Morphology and its Applications to Image and Signal Processing V (ISMM2000)*, J. Goutsias, L. Vincent, D.S. Bloomberg (Eds.), pp. 129–138, Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[11] T. Dandekar, P. Argos, "Potential of genetic algorithms in protein folding and protein engineering simulations", *Protein Engineering 5(7) (1992)*, pp. 637–645.

[12] C. Harris, B. Buxton, "Evolving edge detectors", *Research Note RN/96/3, University College London, Dept. of Computer Science, London (1996)*.

[13] A. Teller, M. Veloso, "A controlled experiment: Evolution for learning difficult image classification", *Proc. 7th Portuguese Conf. on Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 990, Springer-Verlag, Berlin (1995).

[14] R. Poli, S. Cagoni, "Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement", *Genetic Programming 1997: Proc. 2nd Annual Conference on Genetic Programming* Koza, J.R., et al. (Eds.), Morgan Kaufmann, San Francisco (1997).

[15] J.M. Daida, J.D. Hommes, T.F. Bersano-Begey, S.J. Ross, J.F. Vesecky, "Algorithm discovery using the genetic programming paradigm: Extracting low-contrast curvilinear features from SAR images of arctic ice", in *Advances in Genetic Programming 2, chap. 21*, P.J. Angeline, K.E. Kinnear, (Eds.), MIT Press, Cambridge MA, (1996).

[16] J.R. Koza, *Genetic Programming : On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*, MIT Press, Cambridge, MA, (1992).

[17] K.I. Laws, "Texture energy measures", *Proc. Image Understanding Workshop (1979)*, pp. 47–51.

[18] M. Pietikainen, A. Rosenfeld, L.S. Davis, "Experiments with Texture Classification using Averages of Local Pattern Matches", *IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-13, No. 3. (1983)* pp. 421–426.

[19] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press (1995), pp. 105–112.

[20] http://www.rsinc.com

[21] http://www.rsinc.com/Envi/tut2.cfm

[22] J.A. Richards, X. Jia, "Remote Sensing Digital Image Analysis", Springer-Verlag, 1999.

[23] F.A. Kruse, A.B. Lefkoff, J.B. Boardman, K.B. Heidebrecht, A.T. Shapiro, P.J. Barloon, A.F.H. Goetz, "The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data", in *Remote Sensing of Environment, Vol. 44, 1993*, pp. 145–163.

[24] A.S. Mazer, M. Martin, M. Lee, J.E. Solomon, "Image Processing Software for Imaging Spectrometry Analysis", in *Remote Sensing of Environment, Vol. 24, No. 1, 1988*, pp. 201–210.

[25] http://makalu.jpl.nasa.gov

[26] P.G. Weber, B.C. Brock, A.J. Garrett, B.W. Smith, C.C. Borel, W.B. Clodius, S.C. Bender, R.R. Kay, M.L. Decker, "Multi-spectral Thermal Imager mission overview", *Proc. SPIE 3750 (1999)*, pp. 340–346.

Neal R. Harvey received a B.Eng. (Hons.) degree in Mechanical Engineering from the University of Hertfordshire, UK in 1989, a M.Sc. in Information Technology Systems from the University of Strathclyde, UK in 1992 and a Ph.D. in Non-linear Digital Image Processing from the University of Strathclyde in 1997. He spent over 6 years as a Research Fellow in the Signal Processing Division of the University of Strathclyde's Dept. of Electronic and Electrical Engineering. In April 1999, he took up a post-doctoral research associate position in the Space and Remote Sensing Sciences Group of the Nonproliferation and International Security Division at the Los Alamos National Laboratory, NM, USA, where he is currently employed. His research interests include nonlinear digital filters, optimization techniques, machine learning, image classification, remote sensing and film and video restoration.